



# **Test Driven Development**

### **Mock frameworks**



1





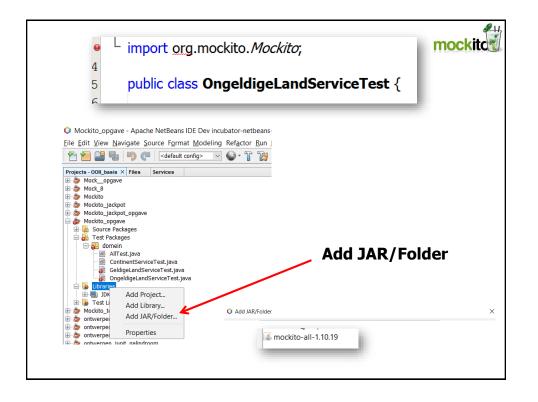
## **Mock frameworks**

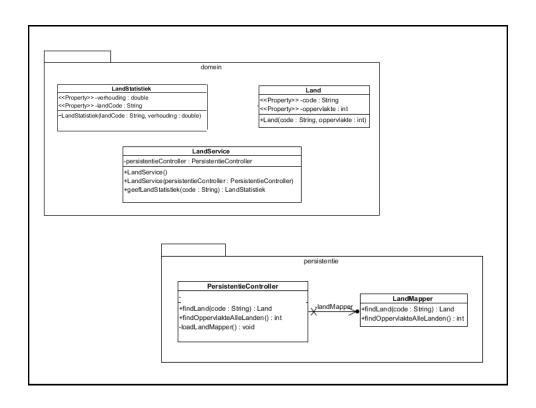
Je hoeft een Mock klasse niet zelf te schrijven.

Er bestaan Mock frameworks die Mock klassen voor je maken:

- Java: JMock, EasyMock, Mockito, ...
- .net: Typemock, Rhino Mocks, Moq, ...







#### 3.1 Mock trainen



```
@Test
   public void landBestaatNiet() {
        final String CODE_GEEN_LAND = "GEEN_LAND";

        Mockito.when(
   persistentieControllerDummy.findLand(CODE_GEEN_LAND)).
        thenReturn(null);

        Mockito.when(
   persistentieControllerDummy.findOppervlakteAlleLanden()).
        thenReturn(100);
```

```
3.2 Na de test verificeren of de mock goed opgeroepen werd

@Test

public void landBestaatNiet() {
    final String CODE_GEEN_LAND = "GEEN_LAND";
    Mockito.when(
        persistentieControllerDummy.
            findLand(CODE_GEEN_LAND)).thenReturn(null);
    Mockito.when(persistentieControllerDummy.
            findOppervlakteAlleLanden()).thenReturn(100);

Assert.assertNull(
    landService.geefLandStatistiek(CODE_GEEN_LAND));

Mockito.verify(persistentieControllerDummy).
    findLand(CODE_GEEN_LAND);

Mockito.verify(persistentieControllerDummy,
            Mockito.verify(persistentieControllerDummy,
            Mockito.times(0)).findOppervlakteAlleLanden();
```

```
We voeren onze verwerking uit:
    landService.geefLandStatistiek(CODE_GEEN_LAND)

Vervolgens kunnen we door 'verify' testen indien de methode findLand, met argument CODE_GEEN_LAND, één keer, in de dummy, werd opgeroepen.

Mockito.verify(persistentieControllerDummy).
    findLand(CODE_GEEN_LAND);

    is hetzelfde als

Mockito.verify(
    persistentieControllerDummy, Mockito.times(1)).
findLand(CODE_GEEN_LAND);
```

#### 4. Speciale gevallen testen

```
@Test(expected = IllegalArgumentException.class)
public void legeCode() {
    landService.geefLandStatistiek("");
}

@Test(expected = IllegalArgumentException.class)
public void spaties() {
    landService.geefLandStatistiek(" ");
}

@Test(expected = IllegalArgumentException.class)
public void nullCode() {
    landService.geefLandStatistiek(null);
}
```

```
package domein;//Test Packages
import ...
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;
import org.mockito.Mock;
import org.mockito.Mockito;
import org.mockito.Mockito;
import org.mockito.MockitoAnnotations;

@RunWith(value = Parameterized.class)
public class GeldigeLandServiceTest {
```

```
1. De mock klasse, het mock object aanmaken
en de mock injecteren in het te testen object
...

@RunWith(value = Parameterized.class)
public class GeldigeLandServiceTest {

private LandService landService;

private PersistentieController persistentieControllerDummy;

@Before
public void before()
{

persistentieControllerDummy =
    Mockito.mock(PersistentieController.class);
landService = new
    LandService(persistentieControllerDummy);
}
```

```
private String landCode;
                                                        mockito
private final int landOppervlakte;
private double verwachteResultaat;
private static final int oppervlakte = 110;
public GeldigeLandServiceTest(String landCode,
    int landOppervlakte, double verwachteResultaat)
  this.landCode = landCode;
  this.landOppervlakte = landOppervlakte;
  this.verwachteResultaat = verwachteResultaat;
}
@Parameters
public static Collection<Object[]> getTestParameters() {
  return Arrays.asList(
      new Object[][]{
          {"BE", 10, 0.1}
         , {"NL", 22, 0.2}
         , {"DE", 78, 0.7}
  );
```

```
@Test
public void testGeefLandStatistiekScenario() {

   Mockito.when(
   persistentieControllerDummy.findLand(landCode))
   .thenReturn(new Land(landCode, landOppervlakte));

   Mockito.when(
   persistentieControllerDummy.findOppervlakteAlleLanden())
   .thenReturn(oppervlakte);
```

```
2.2 Na de test verificeren of de mock goed opgeroepen werd
@Test
  public void testGeefLandStatistiekScenario() {
    Mockito.when(
        persistentieControllerDummy.findLand(landCode))
       .thenReturn(new Land(landCode, landOppervlakte));
    Mockito.when(
        persistentieControllerDummy.findOppervlakteAlleLanden())
       .thenReturn(oppervlakte);
    LandStatistiek stat = landService.geefLandStatistiek(landCode);
    Assert.assertEquals(landCode, stat.getLandCode());
    Assert.assertEquals(
            verwachteResultaat, stat.getVerhouding(), 0.01);
    Mockito.verify(persistentieControllerDummy).
             findLand(landCode);
    Mockito.verify(persistentieControllerDummy).
             findOppervlakteAlleLanden();
}
```